J. Symbolic Computation (1989) **8**, 581–588

Symbolic Computation Application for the Design of Linear Multivariable Control Systems

TAKAJI UMENO, SYUICHI YAMASHITA, OSAMI SAITO AND KENICHI ABE

Toyohashi University of Technology, Tempaku Toyohashi Aichi, 440 *Japan*

(*Received* 16 *November* 1987)

The application of symbolic computation to the algebraic design of linear multivariable control systems is presented. A software package for manipulating polynomial and rational function matrices and for solving the linear matrix equations referred to as unilateral and bilateral equations is implemented on the basis of the symbolic manipulation system known as REDUCE. Several basic functions in the package are explained and their usage is demonstrated by using an example of the design of a discrete-time control system.

1. Introduction

Recently, the algebraic approach to the design of linear multivariable control systems has been widely discussed in the literature (see Kucera, 1979; Desoer *et al.*, 1980; Pernebo, 1981; Vidyasagar, 1985). A characteristic of this approach is that it enables us to solve in a unified way a variety of control problems, such as those concerned with stabilising, decoupling, model matching, tracking and regulating. As the approach requires the manipulation of polynomial and rational function matrices, we encounter difficulties in implementing, in a computational way, the algebraic design procedure. For this reason, many numerical algorithms for the computational problems have been developed (see e.g. Cheng & Pearson, 1981; Zak, 1985; Antsaklis, 1986). These algorithms may be useful for the implementation of computer-aided control design systems.

A further computational method for the algebraic design procedures is to make the best use of existing symbolic manipulation systems such as REDUCE, MACSYMA and SCRATCHPAD, etc., and in fact a number of symbolic manipulation methods for the algebraic approach have been reported (see Tanttu & Altonen, 1985; Saito, Kanno & Abe, 1986).

The aim of this paper is to present a software package for the algebraic approach based on the general symbolic manipulation language REDUCE (see Hearn, 1985). The package allows us to calculate the matrix-fraction description and the Smith form over the polynomial ring and to solve two kinds of linear matrix equations, known as unilateral and bilateral matrix equations, over both the polynomial and generalised polynomial rings. Using this package, the major part of the algebraic design procedure can be successfully performed.

0747–7171/89/120581+08 $03.00/0

2. Algebraic Design Theory—A Review

By a transfer function $G^*(\mu)$, we mean a matrix over the field of rational functions in μ and with real coefficients. Here, μ is the differential operator p or the forward shift operator q depending on whether the system is a continuous- or discrete-time system.

A continuous-time transfer function is usually said to be stable if it has no poles in the closed right-half plane of the complex p-plane. Similarly, a discrete-time transfer function is said to be stable if it has no poles outside the open unit disc with centre at the origin of the complex q-plane. A continuous- or discrete-time system is said to be proper if it is finite at infinity and strictly proper if it equals a zero matrix at infinity.

Following Pernebo (1981) we introduce the transformation

$$\lambda = \begin{cases} \dfrac{1-\tau p}{1+\tau p}, & \tau > 0, \quad \text{continuous-time}, \\ q^{-1}, & \qquad\quad \text{discrete-time}, \end{cases} \tag{1}$$

By this transformation, the unstable region including the infinity will be mapped into the interior of the closed unit disc, which we will call Λ.

Define through

$$G(\lambda) = G^*(\mu),$$

where λ and μ are connected via the above transformation. It is easy to see that $G^*(\mu)$ is proper and stable if $G(\lambda)$ has no poles in Λ.

In agreement with Pernebo (1981), we will adopt the name of Λ-generalised polynomials to rational functions with no poles in Λ. It is known that the Λ-generalised polynomials form a ring and a Euclidean domain just as the polynomials do. Denote the ring of all Λ-generalised polynomials by $R_\Lambda[\lambda]$, and let $R[\lambda]$ be the ring of polynomials and $R(\lambda)$ the field of rational functions. It should be noted that $R[\lambda]$ is a subring of $R_\Lambda[\lambda]$.

Let $R^{mn}[\lambda]$, $R^{mn}(\lambda)$ and $R_\Lambda^{mn}[\lambda]$ be the sets of $m \times n$ matrices with entries in $R[\lambda]$, $R(\lambda)$ and $R_\Lambda[\lambda]$, respectively.

Several concepts for polynomial matrices are valid for Λ-generalised polynomial matrices. We will use the names which are used in polynomial matrices, and only these names will be preceded by Λ. For example, a Λ-generalised polynomial matrix will be said to be "Λ-unimodular" if the determinant of the matrix is a unit in $R_\Lambda[\Lambda]$. Details can be found in Pernebo (1981).

Consider now the control system configuration in Fig. 1, where v, u, u_0 and y are the command input, control input, disturbance input and controlled output, respectively. The plant is assumed to be free of hidden modes and the transfer function in μ of the plant is strictly proper, i.e. the transfer function in λ has no poles at -1 for continuous-time systems or at the origin for discrete-time systems.

Let the controller be described as

$$u = C_1(\lambda)v - C_2(\lambda)y. \tag{2}$$

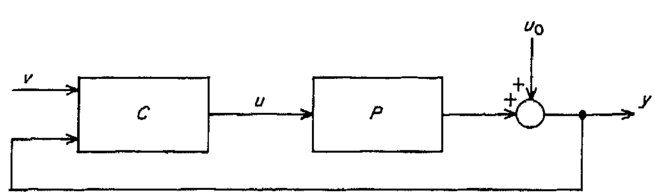

FIG. 1.

This controller must be such that its transfer function in μ is proper and the resultant control system is internally stable.

Define the polynomial matrix-fraction description of the plant $P(\lambda) \in \mathrm{R}^{mn}(\lambda)$:

$$P(\lambda) = D_{p1}^{-1}(\lambda)N_{p1}(\lambda) = N_{p2}(\lambda)D_{p2}^{-1}(\lambda), \tag{3}$$

where (D_{p1}, N_{p1}) is any left and (N_{p2}, D_{p2}) is any right coprime pair of polynomial matrices. Then there exist polynomial matrices X_0, Y_0, X_1 and Y_1 such that

$$D_{p1}X_1 + N_{p1}Y_1 = 1, \quad X_0 D_{p2} + Y_0 N_{p2} = 1. \tag{4}$$

Let

$$C = [C_1 - C_2] = D_c^{-1}[N_\pi - N_f] \tag{5}$$

be a left Λ-coprime, Λ-generalised polynomial matrix-fraction description. The system will be internally stable if and only if the matrix $D_c D_{p2} + N_f N_{p2}$ is Λ-unimodular, or equivalently if and only if D_c and N_f satisfy the unilateral matrix equation

$$D_c D_{p2} + N_f N_{p2} = I. \tag{6}$$

Thus, the set of all stabilising controllers C is represented as follows:

$$C = (X_0 - TN_{p1})^{-1}[N_\pi - (Y_0 + TD_{p1})]. \tag{7}$$

where N_π and T are arbitrary Λ-generalised polynomial matrices.

By a straightforward analysis of Fig. 1, it follows that the transfer function H_{vy} from v to y is

$$H_{vy} = N_{p2} N_\pi \tag{8}$$

and the transfer function from u_0 to y is

$$H_{u0y} = I - N_{p2} N_f. \tag{9}$$

This shows that H_{vy} and H_{u0y} can be independently designed through the choice of two free matrix parameters N_π and T. These two matrix parameters will then be redefined by considering additional constraints on H_{vy} and H_{u0y}. For example, consider the tracking constraint. Let v be described as

$$v = D_v^{-1} N_v v_0, \tag{10}$$

where v_0 is any constant vector, and D_v and N_v are left coprime. The object of tracking is to find a Λ-generalised polynomial N_π such that $H_{vy}v$ is in $\mathrm{R}_\Lambda[\lambda]$. The solution of this problem can be obtained by solving the bilateral equation over $\mathrm{R}_\Lambda[\lambda]$

$$N_{p2} N_\pi + M D_v = I. \tag{11}$$

It is shown that the solution of regulator problems can also be related to the solution of such a bilateral equation. Therefore, the bilateral equation plays the key role in the algebraic approach.

3. New Package on REDUCE for Algebraic Design Procedure

Our package, as well as the most part of REDUCE, is written in RLISP. The main functions of the package are listed in Table 1, which are programmed based on the algorithms in Kucera (1979), arranged for symbolic manipulations. Among these, four significant functions, namely `!*SMITH`, `!*MFD!-L`, `!*UNILATERAL!-L` and `!*BILATERAL`, are explained below. For simplicity, the description is restricted to the ring of $\mathrm{R}[\lambda]$ in this section, and will be extended to $\mathrm{R}_\Lambda[\lambda]$ in the following section.

Table 1.

Function name	Arguments*	Results
!*BILATERAL	A, B, C, var, arb	Solutions of the bilateral equation $AX+YB=C$.
!*DIAG!-MAT	A, var	Diagonalised matrix of A.
!*DIOPHANTINE	a, b, c, var, arb	Solutions of the Diophantine equation $ax+by=c$.
!*DIVISOR!-L(R)	A, B, var	U and V such that $A=BU+V$ ($A=UB+V$), where $\deg V<\deg B$.
!*EXTRACTION!-L(R)	C, G, var, arb	Left(Right) extraction, C' such that $C=GC'$ ($C=C'G$).
!*GCL(R)D	A, B, var	The Left(Right) greatest common divisor of A and B.
!*L(R)STRUCT	H, var	Left(Right) structure matrix of H.
!*MFD!-L(R)	F, var	Left(Right) coprime fraction of F.
!*SMITH	A, var	Smith form of A.
!*STABLEP	a, var	If a has no zeros in Λ then T else NIL.
!*UNILATERAL!-L(R)	A, B, C, var, arb	Solutions of the unilateral equation $AX+BY=C$ ($AX+YB=C$).

* F = rational matrix; A, B, C and G = polynomial matrices; $H = \Lambda$-generalised polynomial matrix; a, b and c = polynomials; var = main variable; arb = parameter for the general solutions.

!*SMITH: Let A belong to $R^{mn}[\lambda]$. This function produces the Smith form A_S of A and unimodular matrices $U\in R^{mm}[\lambda]$ and $V\in R^{nn}[\lambda]$ such that $A_S=UAV$.

!*MFD!-L: Let H belong to $R^{mn}(\lambda)$. This function finds a left coprime matrix fraction description $H=A^{-1}B$, where $A\in R^{mm}[\lambda]$ and $B\in R^{mn}[\lambda]$.

!*UNILATERAL!-L: Let A, B and C belong to $R^{mp}[\lambda]$, $R^{mq}[\lambda]$ and $R^{mn}[\lambda]$, respectively. This function finds the general solutions $X\in R^{pn}[\lambda]$ and $Y\in R^{qn}[\lambda]$ satisfying the left unilateral equation $AX+BY=C$. The algorithm is summarised as follows.

(1) Find a GCLD $G\in R^{mm}[\lambda]$ of A and B, and a unimodular matrix $U\in R^{p+q\ p+q}[\lambda]$ such that $[A\ B]U=[G\ 0]$, where G is a lower triangular matrix referred to as Hermite normal form. Let r denote the rank of G.

(2) Partition U into four submatrices as below:

$$U=\begin{array}{c c} & \begin{array}{cc} r & p+q-r \end{array} \\ \begin{array}{c} p \\ q \end{array} & \begin{bmatrix} P & R \\ Q & S \end{bmatrix} \end{array}. \tag{12}$$

(3) By calling function !*EXTRACTION!-L, find $C_1\in R^{mn}[\lambda]$ such that $C=GC_1$. If C_1 does not exist, then no solutions exist.

(4) If $r=p+q$ then $X=PC_1$, $Y=QC_1$. If $r<p+q$ then $X=PC_1+RT$, $Y=QC_1+ST$, where $T\in R^{p+q-r}[\lambda]$ is an arbitrary polynomial matrix.

!*BILATERAL: Let A, B and C belong to $R^{mp}[\lambda]$, $R^{qn}[\lambda]$ and $R^{mn}[\lambda]$, respectively. And suppose the rank of A and B are r and s, respectively. This function finds the general solutions $X\in R^{pn}[\lambda]$ and $Y\in R^{mq}[\lambda]$ satisfying the bilateral equation $AX+YB=C$. The algorithm is summarised as follows.

(1) By calling function !*SMITH, find the Smith forms of the following two matrices

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}, \quad \begin{bmatrix} A & C \\ 0 & B \end{bmatrix}.$$

If these Smith forms are not equal to each other, no solutions exist.

```
!*MFD!-R(P,D);  <----------Function call. P is the transfer matrix, and D indicates that the main
                           variable of P is "D".
H = BR*AR**(-1) = B2*A2**(-1)
                  A2 = AR*G2**(-1)                                                  menu
                  B2 = BR*G2**(-1)
   P2*A2+Q2*B2 = I
   R2*AR+S2*BR = 0

 (0)    H   ; (1) AR ; (2) BR ; (3) A2 ; (4) B2 ;
 (5)   G2   ; (6) P2 ; (7) Q2 ; (8) R2 ; (9) S2 ;
(10)   END  ;

Input the number.
4;  <--------------Input the number 4 to get "numerator matrix" of P.
B2:=

| - D^3 - 5*D^2 - 7*D - 3   *(1,2) |
| 2*( - D^2 - 2*D - 1)        0    |   <---Matrix can be displayed in natural form.

*(1,2):=( - D^4 - 4*D^3 - 2*D^2 + 4*D + 3)/(2*(D^2 + 2*D - 3))

Define this matrix. ($: non define)
NP2;  <-------------Define above matrix as NP2.
```

Fig. 2. Operation in the function !*MFD!-R.

(2) Find the Smith forms A' and B' of A and B, respectively, and corresponding unimodular matrices U_a, V_a, U_b and V_b such that $A'=U_aAV_a$ and $B'=U_bBV_b$. Let $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_s$ be the invariant polynomials of A and B, respectively.

(3) Let $C'=U_aCV_b$. Then the original equation is transformed into $A'X'+Y'B'=C'$, where $X'=V_a^{-1}XV_b$ and $Y'=U_aYU_b^{-1}$.

(4) Find x'_{ij} and y'_{ij} that satisfy the following equations.

   (i) $a_ix'_{ij}+y'_{ij}b_j=c'_{ij}$, $\quad 1\leqslant i\leqslant r$ and $1\leqslant j\leqslant s$,
   (ii) $a_ix'_{ij}=c'_{ij}$, $\quad 1\leqslant i\leqslant r$ and $s+1\leqslant j\leqslant n$,
   (iii) $y'_{ij}b_j=c'_{ij}$, $\quad r+1\leqslant i\leqslant m$ and $1\leqslant j\leqslant s$.

(5) The rest of the entries of X' and Y' are set to arbitrary polynomials.

(6) $X=V_aX'V_b^{-1}$ and $Y=U_a^{-1}Y'U_b$.

The polynomial equation (i) in step (4) is solved by calling function !*DIOPHANTINE, which can find the general solution of the equation.

As shown in Fig. 2, we adopt the menu system for each function so that a user can select any desired result by specifying a corresponding number. We expect the menu system will be helpful to users with less knowledge of LISP or REDUCE.

Finally, it is noted that the functions of our package include those of the package developed by Tanttu & Altonen (1985).

4. The Solutions of Λ-generalised Polynomial Matrices

Since $R_\Lambda[\lambda]$ is a Euclidean domain, most results for polynomial matrices are also valid in appropriately modified form, for Λ-generalised polynomial matrices (see Pernebo, 1981). This means that the algorithms for the ring $R[\lambda]$ in the previous section can be extended easily to those for the ring $R_\Lambda[\lambda]$. Here, we show how to extend the functions !*UNILATERAL!-L(R) and !*BILATERAL so as to enable us to derive Λ-generalised polynomial matrix solutions.

The Λ-generalised polynomial matrix solutions of the unilateral equation can be obtained by the algorithm of !*EXTRACTION!-L with step (3) replaced by (3′):(3′). By calling function !*EXTRACTION!-L, find $C_1 \in \mathrm{R}^{mn}{}_{\Lambda}[\lambda]$ such that $C = GC_1$. If C_1 does not exist, then no solutions exist. In this process, function !*STABLEP is called in !*EXTRACTION!-L and determines whether all entries of C_1 belong to $\mathrm{R}_{\Lambda}[\lambda]$ or not.

The function !*DIOPHANTINE is also employed to obtain the Λ-generalised polynomial matrix solutions to the bilateral equation. Let a, b and c belong to $\mathrm{R}[\lambda]$. Then this function is able to find the solutions x, $y \in \mathrm{R}_{\Lambda}[\lambda]$ satisfying the equation $ax + by = c$ if and only if $c/g \in \mathrm{R}[\lambda]$, where g is the greatest common divisor of a and b. Solutions x and y are obtained as follows:

$$x = (c/g)p + (b/g)t, \qquad y = (c/g)q + (a/g)t, \tag{13}$$

where p and q are such that $ap + bq = g$, and t is any element in $\mathrm{R}_{\Lambda}[\lambda]$. Therefore, it is clear that the function !*BILATERAL can solve bilateral equations over $\mathrm{R}_{\Lambda}[\lambda]$.

The user has to choose between the ring $\mathrm{R}[\lambda]$ and the ring $\mathrm{R}_{\Lambda}[\lambda]$ for the use of these functions. To do so, an optional switch "LAMBDA" is newly defined in the REDUCE system. The switch is turned on and off by commands ON and OFF, respectively. Hence, the command "ON LAMBDA;", in a REDUCE session, causes the system to derive solutions of Λ-generalised polynomial matrices.

5. Example

In this section, we illustrate the performance of our package with an example of the design of a discrete-time control system.

Consider the following plants

$$P = \begin{bmatrix} \dfrac{-(d+1)^2}{2(d-1)} & 0 \\ \dfrac{-(d+1)^2}{(d-1)(d+3)} & \dfrac{(d+1)^2}{2(d+3)} \end{bmatrix}, \tag{14}$$

where symbol d means λ.

The first step of the design procedure is to parameterise a class of the stabilising controllers. A right coprime fraction description of P can be obtained by !*MFD!-R. When this function is called, the menu is displayed as shown in Fig. 2. Here, input the number "4", then N_{p2} of eqn (3) is derived. Then, the same menu appears again, we input the numbers 3, 6 and 7 successively to obtain D_{p2}, X_0 and Y_0. A pair of (X_0, Y_0) is shown below

$$X_0 := \begin{vmatrix} (-1)/8 & (D-1)/16 \\ 0 & 1/2 \end{vmatrix}, \qquad Y_0 := \begin{vmatrix} 0 & (-8) \\ 0 & 0 \end{vmatrix},$$

Then, by using these matrices, we easily construct the stabilising controllers (eqn (7)). Of course, the matrices D_{p1} and N_{p1} of eqn (7) are to be derived by function !*MFD!-L. Next, let us solve the tracking problem for the following command input

$$v = \begin{bmatrix} \dfrac{(d+1)^2}{(d-1)^2} & \dfrac{-(d+1)}{d-1} & 0 \\ 0 & 0 & \dfrac{-(d+1)}{d-1} \end{bmatrix} v_0. \tag{15}$$

```
!*BILATERAL(NP2,DV,I2,D,R);  <----------Equation to be solved is
                                         NP2*NPI+M*DV = I2.
A*X+Y*B = C  <-------X and Y correspond to NPI and M, respectively.

(0) A  ;(1)  B  ;(2)  C  ;(3)  X  ;(4)  Y  ;
(5) END ;

Input the number.
3:  <--------------Input the number 3 to get the solution NPI.
X:=

|( - 4*D^2*R2 + 8*D*R2 + D - 4*R2 - 2)/4  R1*( -D + 1)          |
|R4*(D^2 - 2*D+1)                         (4*D*R3 - 4*R3 +1)/4  |

Define this matrix. ($: non define)
NPI;  <-------------Define above matrix as NPI.
```

Fig. 3. Operation in the function !*BILATERAL.

Using !*MFD!-L, we can find a left coprime fraction description of v. After that, N_π satisfying eqn (11) can be derived by !*BILATERAL. The result is shown in Fig. 3, where NPI represents N_π and R1, R2, R3 and R4 represent arbitrary Λ-generalised polynomials.

In the above process, we have determined the matrices XO, YO, NP1, DP1 and N_π necessary for the construction of the tracking controllers.

Finally, we find the parameterisation for the tracking controller:

$$C := \begin{vmatrix} *(1,1) & ((4*D*R3+32*R1-4*R3+1)*(D-1))/4 & 0 & (-1) \\ 2*(D-1)^2*R4 & (4*D*R3-4*R3+1)/2 & 0 & 0 \end{vmatrix},$$

$$*(1,1) := -((16*R2-3*R4+2)*D-(8*R2-3*R4)*D^2-D^3*R4-8*R2 \\ +R4-4).$$

We note that it required about 17 cpu seconds to complete the above design procedure by DEC machine, micro VAX-II. In particular, the function !*BILATERAL consumes 10 s in solving the equation. Although the matrix size that can be treated on our package depends on the computer system, we are restricted to 5×5 matrices, because the general solutions of the matrix equations would be very complicated expressions.

6. Conclusions

It has been shown that the symbolic manipulation system REDUCE can be successfully applied to the algebraic design procedure of linear multivariable control systems. Several functions were newly added to REDUCE, and their usage was demonstrated by solving the stabilising and tracking problems.

We remark finally that our package will serve as both educational and research tools in the algebraic theory of linear multivariable control systems.

References

Antsaklis, P. J. (1986). Proper stable matrix factorizations and internal system descriptions. *IEEE Trans. Automat. Contr.*, **AC-31**, 634–638.

Cheng, L., Pearson, J. B. (1981). Synthesis of linear multivariable regulators. *IEEE Trans. Automat. Contr.* **AC-26**, 194–202.

Desoer, C. A., Liu, R. W., Murray, J. J., Saeks, R. (1980). Feedback system design: the fractional approach to analysis and synthesis. *IEEE Trans. Automat. Contr.*, **AC-25**, 401–412.

Hearn, A. C. (1985). *REDUCE User's Manual (Version 3.2).* Rand Publication.
Kucera, V. (1979). *Discrete Linear Control: The Polynomial Equation Approach.* New York: Wiley.
Pernebo, L. (1981). An algebraic theory for the design of controllers for linear multivariable systems. *IEEE Trans. Automat. Contr.*, **AC-26**, 171–174.
Saito, O., Kanno, M., Abe, K. (1988). Symbolic manipulation CAD of control engineering. *Int. J. Contr.*, to be published.
Tanttu, J. T., Altonen, M. (1985). Max—a program system for symbolic manipulation of polynomial matrices. *Proc. 3rd IFAC/IFIP Symp. on Computer Aided Design in Control and Engineering Systems.* Denmark: Lyngby.
Vidyasagar, M. (1985). *Control System Synthesis: A Factorization Approach.* Reading, MA: MIT press.
Zak, S. H. (1985). On the polynomial matrix equation $AX+YB = C$. *IEEE Trans. Automat. Contr.*, **AC-30,** 1240–1242.